

Каменева Анастасия Евгеньевна

магистрант

Горбунова Алина Викторовна

студентка

ФГБОУ ВПО «Магнитогорский государственный
технический университет им. Г.И. Носова»
г. Магнитогорск, Челябинская область

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ КОМПОНЕНТА WEBVIEW В WINDOWS PHONE 8.1

***Аннотация:** в данной статье рассматривается компонент *WebView* платформы *WP8.1* и такие специфические особенности его использования, как регулирование высоты, выполнение *Javascript* и загрузка локальных ресурсов.*

***Ключевые слова:** программирование, NET, Windows Phone 8, *WebView*.*

Компонент, предназначенный для отображения HTML-содержимого – важная часть любого фреймворка. В.NET Framework для Windows Phone 8.1 этот компонент реализован классом *WebView*.

Отметим некоторые основные методы этого класса:

– *Navigate (Uri Source)* – загружает HTML-содержимое, расположенный по указанному *Uri*;

– *NavigateToLocalStreamUri (Uri source, IUriToStreamResolver streamResolver)* – загружает локальное веб-содержимое по указанному *URI*, используя *IUriToStreamResolver*;

– *NavigateToString(string Source)* – загружает указанное в строке HTML-содержимое в новый документ;

– *InvokeScriptAsync(string scriptName, string[] arguments)* – выполняет заданную функцию скрипта из текущего загруженного HTML с конкретными аргументами.

Рассмотрим некоторые особенности, которые возникают при использовании этого компонента.

Нужно отметить, что `WebView` не является подклассом `Control` и, таким образом, не имеет шаблона элемента управления. Область отображения ограничена свойствами `Width` и `Height`. Однако, при присвоении свойству `Height` значения `Auto`, размер элемента не будет отрегулирован по высоте HTML-содержимого – высота элемента будет нулевой, и на экране он не отобразится. Это также касается размещения `WebView` в панели `Grid` с высотой строки, равной `Auto` или `*`. Чтобы избежать подобной проблемы, необходимо, во-первых, задавать ненулевую минимальную высоту `WebView` `MinHeight`, во-вторых, для более точной регулировки размера, либо изначально задавать высоту в пикселях (свойство `WebView.Height`), либо вызывать из HTML-содержимого скрипт, который рассчитывает высоту контента и присваивать `WebView` высоту после загрузки содержимого.

Также отметим, что `javascript` включен по умолчанию. Что касается оповещений (`javascript alert`), `WebView` их показывать не будет. Вывести их на экран можно с помощью вызова функции `WebView.InvokeScriptAsync`. Размещаемая HTML-страница может инициировать событие `ScriptNotify`, когда страница вызывает `window.external.notify` и передает параметр строки. В событии `ScriptNotify` перехватывается содержимое оповещения и, как правило, выводится на экран в виде стандартного диалога.

Отдельно рассмотрим механизм загрузки содержимого из локальных потоков. Метод `NavigateToLocalStreamUri` используется для загрузки локального содержимого, не обрабатываемого методом `NavigateToString`.

Чтобы использовать метод `NavigateToLocalStreamUri`, необходимо передать реализацию `IUriToStreamResolver`, преобразующую шаблон URI, в поток содержимого. Это можно сделать, чтобы предоставить содержимое для всех ресурсов, используемых веб-страницей или рядом страниц. Например, этот метод можно использовать для отображения содержимого, сохраненного в локальной файловой системе в виде зашифрованных файлов или в `cab`-пакетах. Если запрашивается содержимое, можно использовать реализацию `IUriToStreamResolver` для его расшифровки на лету.

Интерфейс `IUriToStreamResolver` имеет один метод `UriToStreamAsync`, который принимает URI и возвращает поток. Универсальный код ресурса (URI) для перехвата должен быть в формате `ms-local-stream://appname_KEY/folder/file` или `folder/file`, где KEY определяет распознаватель.

Однако, `IUriToStreamResolver` позволяет загружать в `WebView` только потоки, полученные из файлов приложения, свойство `Build Action` у которых установлено как `Content`, т.е. содержимое этих файлов должно быть загружено через интерфейс работы с файловой системой. Практика показала, что загрузить поток, например, из встроенных ресурсов приложения возможно и ошибок при выполнении не возникнет, однако `WebView` отображать содержимое этого потока не будет.

Список литературы

1. Пугачев С.В. Разработка приложений для Windows 8 на языке C# / С.В. Пугачев, А.М. Шериев, К.А. Кичинский. – СПб.: БХВ-Петербург, 2013. – 416 с.
2. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft.NET Framework 2.0 на языке C#. Мастер-класс / Пер. с англ. – 2-е изд., исправ. – М.: Русская Редакция; СПб.: Питер, 2008. – 656 с.